**به نام خدا**

# تمرین ۲، گزارش مربوط به دیتاست ۱

**علیرضا نجاتی، ۹۸۲۲۲۱۰۴**

**مقدمه:**

این دیتاست مربوط به اطلاعات انواع مختلف تلفن همراه می باشد که در چهار کلاس قیمتی مختلف طبقه بندی شده اند. با توجه به این داده ها می توان به ارتباطی که هر یک از ویژگی های تلفن همراه با کلاس قیمتی خود دارد پی برد. در این تمرین قصد داریم با روش های انتخاب ویژگی (Feature selection) از جمله روش پیشرو (Forward) و پسرو (Backward) آشنا شویم، آن ها را پیاده سازی کنیم و تفاوت مدل آموزش داده شده با ویژگی های انتخاب شده توسط این دو روش را بررسی کنیم. سپس از روش PCA برای کاهش بعد داده ها استفاده کرده و نتیجه مدل را گزارش می کنیم. در ادامه به مهندسی ویژگی (Feature engineering) که یکی از بخش های مهم در فرایند های یادگیری ماشین است، می پردازیم. پس از انجام هر یک از روش های مهندسی ویژگی مدل SVM می سازیم و نتیجه مدل را گزارش می کنیم.

**توضیحات:**

۶ سلول ابتدایی مربوط به اتصال به کگل، دریافت دیتاست، دسترسی کتابخانه های مورد نیاز و ساخت دیتافریم از روی دیتاست می باشد.

۱- **پیش پردازش داده ها:**

داده ی پوچ در دیتاست ما وجود ندارد. با توجه به ناچیز بودن تعداد داده های پرت و همچنین زیاد نبودن حجم کل داده ها، از آن ها چشم پوشی کردیم.

۲- **تسک اول :**

در این تسک قرار است به وسیله روش انتخاب ویژگی پیشرو تعدادی از فیچر ها را انتخاب کنیم. معیاری که بوسیله آن فیچر ها را با هم مقایسه میکنیم AUC است. ابتدا لیبل کلاس های ارزان یعنی ۰و۱ را باهم ادغام و کلاس های گران را نیز با هم ادغام میکنیم. پس از جدا سازی ستون هدف از بقیه ستون ها با استفاده از روش Standard Scaler داده ها را هم مقیاس می کنیم. سپس داده های آموزش و تست را جدا کرده و مدل Logestic Regression ساخته و با استفاده از این مدل بر روی داده های تست پیش بینی را انجام داده و AUC را گزارش میکنیم که مقدار آن برای داده ها قبل از انتخاب ویژگی برابر با ۰,۹۸۲۵ شده است.

در نهایت روش انتخاب ویژگی پیشرو را پیاده سازی می کنیم. ابتدا یک آرایه خالی در نظر می گیریم در هر مرحله یک فیچر به آرایه اضافه می کنیم و برای دیتافریم حاصل مقدار AUC را بدست می آوریم و بهترین AUC بدست آمده را نگه می داریم و آن ویژگی را به آرایه نهایی اضافه می کنیم. پس از انجام این تابع نتیجه بدست آمده شامل ۸ فیچر زیر است.

['ram', 'battery_power', 'px_height', 'px_width', 'touch_screen', 'mobile_wt', 'n_cores']

**۳- تسک دوم :**

در این بخش مدل Logestic Regression را بر روی فیچر های انتخاب شده در بخش قبل، پیاده سازی می کنیم. مقادیر Precision برابر ۰,۹۹۰۱ ، Recall برابر ۰,۹۹ و F1_score برابر ۰,۹۸۹۹ بدست آمد.

**۴- تسک سوم :**

در این بخش با استفاده از روش PCA بُعد را به همان تعداد فیچر های انتخاب شده در بخش اول کاهش می دهیم.

**۵- تسک چهارم :**

در این بخش معیار های Precision، Recall و F1_score برای داده های کاهش بعد داده شده در بخش قبل گزارش می کنیم که به ترتیب برابر با ۰,۶۸۰۱، ۰,۶۸ و ۰,۶۷۹۹ هستند.

**۶- تسک ششم:**

برخی از روش های مهندسی ویژگی را در این بخش پیاده سازی می کنیم.

اولین روش، Binning بر روی ویژگی battety_power است. برای این کار موبایل های دارای توان باتری ۵۰۱ تا ۹۰۰ را با برچسب bad ، موبایل های دارای توان باتری ۹۰۱ تا ۱۶۰۰ را با برچسب normal و موبایل های دارای توان باتری ۱۶۰۱ تا ۲۰۰۰ را با برچسب good برچسب گذاری می کنیم.

در روش دوم قرار است داده های غیر عددی را به داده های عددی تبدیل کنیم که در این دیتاست داده تنها فیچر battery power غیر عددی است که خودمان آن را بدین شکل تبدیل کردیم. با متد One hot encoding این کار را انجام می دهیم.

*"بسیاری از الگوریتم های یادگیری ماشین توانایی کار با داده های غیر عددی را ندارند و نیاز دارند همه ی ورودی هایشان داده های عددی باشند و خروجی های عددی نیز ارائه کنند. بنابر این باید داده های غیر عددی را به داده های عددی تبدیل کنیم. یکی از روش ها برای این کار* ***one hot encoding*** *می*

*باشد. این روش داده ها را بسیار کاربردی تر و خوانا تر می کند و نرمال سازی یا استاندارد سازی داده ها را آسان تر می کند."*

در بخش بعدی میخواهیم داده ها را با تبدیلات لگاریتمی یا نمایی عوض کنیم. *"به طور کلی زمانی از این تبدیلات استفاده می کنیم که یا در توزیع داده ها چولگی داشته باشیم ویا اختلاف مقادیر داده های در یک ستون بسیار زیاد باشد."*

ابتدا برای هر فیچر مقدار چولگی را با استفاده از تابع skew بدست می آوریم و از بین آن ها دو فیچر fc و sc_w انتخاب کرده و تبدیل لگاریتمی را روی آن اعمال می کنیم. نمودار این دو فیچر را قبل و بعد از این تبدیل رسم می کنیم.

در بخش آخر نیز یک فیچر جدید تحت عنوان مساحت (sc_area) به وسیله دو فیچر sc_w و sc_h می سازیم.

**۷- تسک هفتم:**

در این بخش می خواهیم مدل SVM را پیاده سازی کنیم. ابتدا این کار را بر روی داده های اصلی انجام می دهیم. دقت در این حالت برابر با ۰,۹۶ است. سپس مدل را بر روی داده های بدست آمده پس از انجام عمل binning و one hot encoding پیاده سازی می کنیم. دقت برای این مدل برابر است با ۰,۸۲۷۵. سپس مدل را بر روی داده های بدست آمده پس از انجام تبدیل لگاریتمی پیاده سازی می کنیم. دقت برای این مدل برابر است با ۰,۹۶. سپس مدل را بر روی داده های بدست آمده پس از ساخت فیچر جدید پیاده سازی می کنیم. دقت برای این مدل برابر است با ۰,۹۶. در نهایت مدل را بر روی داده های بدست آمده پس از اجرای همه روش های مهندسی ویژگی گفته شده پیاده سازی می کنیم که دقت در این حالت برابر با ۰,۸۲۲۵ است.

**۸- تسک هشتم :**

*Bootstrapping یک روش نمونه گیری مجدد با جایگذاری است. این روش با طراحی مکرر نمونه ها از داده های اصلی به وسیله جایگذاری، برای تخمین پارامتر جمعیت به کار می رود. در این حالت ممکن است داده های بوت استرپ شده حاوی دادی تکراری از داده های اصلی باشند و بعضی از داده ها از آن حذف شوند. تفاوت آن با روش Cross validation این است که روش cross validation روش نمونه گیری مجدد بدون جایگذاری است. بدین صورت که داده ها به K قسمت تقسیم شده و هر بار یک قسمت از داده ها برای تست کنار گذاشته می شوند. به اصطلاح به این روش K-fold cross validation گفته می شود. روش Bootstrapping برای ساختن مدل تجمیعی و یا تخمین پارامتر های آماری از روی یک مشاهده برای کل جامعه به کار می رود.*

**۹- تسک نهم :**

*روش 5 x 2-cross validation شامل ۵ تکرار یک ۲-fold cross validation است. عدد دو بخاطر این است که اطمینان مشاهده هر داده فقط در مجموعه آموزش با فقط در مجموعه تست حاصل شود. این روش برای تخمین عملکرد مدل، خطای مدل و واریانس به کار می رود و میتوان از آن برای مقایسه مدل های مختلف استفاده کرد.*

**۱۰- تسک دهم :**

*به طور کلی نمیتوان با استفاده از این روش بهترین مرتبه مدل را پیدا کرد. زیرا هرچقدر مرتبه مدل کمتر باشد مقدار خطا و بایاس ما افزایش پیدا می کند و هرچقدر بیشتر باشد مقدار واریانس ما افزایش پیدا می کند. در دنیای واقعی دیتاست ها خیلی تمیز نیستند و دارای نویز بسیاری هستند. برای این دیتاست ها روش elbow ممکن است روش کارامدی برای پیدا کردن مرتبه مدل باشد اما اگر دیتاستی داشته باشیم که نویز نداشته باشد دیگر این روش نمی تواند بهترین مرتبه را به ما ارائه کند.*

**۱۱- تسک امتیازی دوم :**

*برای مقایسه مدل های یادگیری ماشین با یکدیگر معمولا از روش های نمونه گیری مجدد استفاده می شود (مانند k-fold cross validation). اما مشکلی که این روش ها دارند این است که ما نمیتوانیم مطمعن باشیم که اختلاف نتیجه حاصل از این روش ها واقعی است یا بدلیل برخی از مسائل آماری است. برای اینکه با این مشکل مواجه نباشیم از Statistical significance tests استفاده می کنیم. این تست ها احتمال اینکه نمره ها ازروی داده هایی که از یک توزیع یکسان بدست آمده اند را محاسبه می کنند. روش های McNemar's test و 5 x 2 cross validation نمونه های از این تست ها هستند.*

**۱۲- تسک امتیازی سوم :**

*معیار MCC یک معیار برای ارزیابی طبقه بند های چند کلاسه یا باینری است. این معیار کلاس پیش بینی شده و کلاس واقعی را به عنوان دو متغیر در نظر می گیرد و correlation بین آن ها را محاسبه می کند. مقدار MCC همواره بین ۱- و ۱ است که هرچه به ۱ نزدیک تر باشد مدل پیش بینی بهتری انجام داده است و اگر ۰ باشد یعنی به صورت رندوم پیش بینی را انجام داده است.*

**منابع :**


https://www.educative.io/blog/one-hot-encoding

https://medium.com/@kyawsawhtoon/log-transformation-purpose-and-interpretation-9444b4b049c9

-٠٨https://carpentries-incubator.github.io/machine-learning-novice-python/bootstrapping/index.html

https://machinelearningmastery.com/statistical-significance-tests-for-comparing-machine-learning-algorithms/

https://towardsdatascience.com/the-best-classification-metric-youve-never-a٩e٣f٢a٥٠bf٣heard-of-the-matthews-correlation-coefficient-